```
# UCSC Extension
# DBDA.X409.(10) MySQL and Oracle Database for Developers and
Designers
# Assignment 3
# Cheng Fei
# 10/23/2022
#
# A set of functions using Dynamic SQL to process data in studentdb
database.

# Change the current database to studentdb.
USE studentdb;

# @_CREATE_PROCEDURE_
# A database procedure in Dynamic SQL that accepts one input
parameter {student_id and event_id}.
# Use these input parameters to return the category that this
student belongs to.
# Also display the student_id and event_id in the result.
DELIMITER $

DROP PROCEDURE IF EXISTS student_category_by_name_event;

CREATE PROCEDURE student_category_by_name_event(IN in_student_id
INT, IN in_event_id INT)
BEGIN
        # Basic variables declaration.
    DECLARE v_student_id INT;
    DECLARE v_event_id INT;
    DECLARE v_output_table VARCHAR(100); # The title of the data
table.
    # Dynamic SQL statements declaration.
    DECLARE drop_table VARCHAR(1000); # Dynamic SQL statement to
drop the existing table.
    DECLARE add_table VARCHAR(1000); # Dynamic SQL statement to add
selected data to a new table.
    DECLARE show_table VARCHAR(1000); # Dynamic SQL statement to
show selected data.
        BEGIN
                # Basic variables assignment.
                SET @v_student_id = in_student_id;
        SET @v_event_id = in_event_id;
        SET @v_output_table = 'student_category';
        # Dynamic SQL statements initialization.
        # Drop the existing table.
        SET @drop_table = concat('DROP TABLE IF EXISTS ',

@v_output_table);
                PREPARE stmt1 FROM @drop_table;
        EXECUTE stmt1;
        # Add selected data to a new table.
        SET @add_table = concat('CREATE TABLE ',

@v_output_table,
                                ' AS ',
                                'SELECT s.student_id, s.event_id,
g.category ',
                                'FROM score s, grade_event g ',
                                'WHERE s.student_id = ? and
s.event_id = ? ',
                                'and s.event_id = g.event_id');
                PREPARE stmt2 FROM @add_table;
        EXECUTE stmt2 USING @v_student_id, @v_event_id;
        # Show the selected data.
        SET @show_table = concat('SELECT * FROM ',

@v_output_table);
                PREPARE stmt3 FROM @show_table;
        EXECUTE stmt3;
    END;
END$

DELIMITER ;

# @_CREATE_FUNCTION_
# The function takes event_id and student_id as input,
# and finds out the score for a given student with its relevant
event.
# This function use a SUB QUERY to check that
# both student_id and event_id exists in both student and
grade_Event tables.
DELIMITER $

DROP FUNCTION IF EXISTS student_score_event_by_ids;

CREATE FUNCTION student_score_event_by_ids(in_student_id INT,
in_event_id INT)
RETURNS VARCHAR(100) # Return selected data in a table.
READS SQL DATA
BEGIN
        DECLARE out_student_score_event VARCHAR(100);
        BEGIN
                SELECT score INTO out_student_score_event
        FROM score
        WHERE student_id = in_student_id
        and student_id IN (SELECT student_id FROM student)
        and event_id = in_event_id
        and event_id IN (SELECT event_id FROM grade_event);

        RETURN out_student_score_event;
    END;
END$

DELIMITER ;

# Commit changes.
COMMIT;
```